## **APPENDIX**

The following code listing shows one implementation of the conventional VSCALE routine in accordance with ITU (International Telecommunication Union)-T Recommendation G.728 – Annex G.

5

```
; search for maximum positive and negative values in vector
                                movs.w @r4+,y0
                                movx.w @r4+,x1
             pcopy y0,y1.
 10
             pcmp x1,y0
         dct pcopy x1,y0
             pcmp x1,y1
         dcf pcopy x1,y1
                                movx.w @r4+,x1
             pcmp x1,y0
 15
         dct pcopy x1,y0
             pcmp x1,y1
         dcf pcopy x1,y1
                                movx.w @r4+,x1
             pcmp x1,y0
         dct pcopy x1,y0
20
             pcmp x1,y1
         dcf pcopy x1,y1
                                movx.w @r4,x1
ī
             pcmp x1,y0
                                movx.w @r4+r8,x0
恒
         dct pcopy x1,y0
                                movx.w @r4+r8,x0
Œ
             pcmp x1,y1
                                movx.w @r4+r8,x0
25
         dcf pcopy x1,y1
                                movx.w @r4+r8,x0
إيبا
m
             sts
                   y0,r1
             r1,r0
      mov
Ţ
                   y1,r7
             sts
30 mm mm mm mm 35
                   r7,r0
             or
                   r0,r0
             tst
             bt
                   VS_ZERO
             pabs
                   y1,y1
             pclr
                   a0
             pinc
                   a0,a0
                   r6,y0
             lds
             psha
                   #16,y0
             psha
                   a0,y0,a0
 40
             sts
                   y1,r0
             cmp/ge
                          r0,r1
                   vs_pos
             bt/s
                    #0,r0
             mov
 45
                   a0, r3
             sts
                   r3, r3
             neg
             mov
                    r3, r2
             shll
                   r2
 50
             cmp/ge
                          r2, r7
             bf
                    vsloop3
             cmp/gt
                          r7, r3
                   vs_end2
 55
       ;Case 3: maximum negative value still has room for normalization
             .align
       vsloop41:
             shal r7
 60
             cmp/qt
                          r7, r3
             bf/s vsloop41
             add #1,r0
```

```
lds
                  r0,y0
      psha #16,y0
            movs.w @r4+,x1
  5
            psha x1,y0,a0
                              movx.w @r4+,x1
      psha x1,y0,a1
                      movx.w @r4+,x1
                               movs.w a0,@r5+
                               movx.w a1,@r5+
            psha x1,y0,a0
                               movx.w @r4+,x1
 10
      psha x1,y0,a1
                        movx.w @r4+,x1
                               movx.w a0,@r5+
            psha x1,y0,a0
      movx.w al,@r5+
                               movx.w a0,@r5+
 15
            rts
            nop
      ;Case 2: maximum negative value exceeds minimum range vsloop3:
 20
                       r2,r7
            cmp/ge
            bt
                 vs_end2
.align
:II
      vsloop31:
IJ
            shar r7
25
            cmp/ge
                        r2, r7
1
      bf/s vsloop31
1.4
            add
                  #-1,r0
131
            lds
                  r0,y0
            psha #16,y0
            movs.w @r4+,x1
psha x1,y0,a0
psha x1,y0,a1
                               movx.w @r4+,x1
Ħ
                               movx.w @r4+,x1
1
                               movs.w a0,@r5+
35
                               movx.w a1,@r5+
12
            psha x1,y0,a0
                               movx.w @r4+,x1
1
            psha x1,y0,a1
                               movx.w @r4+,x1
                               movx.w a0,@r5+
            psha x1,y0,a0
 40
                               movx.w a1,@r5+
                               movx.w a0,@r5+
            rts
            nop
 45
      ;Case 1: zero input vector
      VS_ZER0:
            pclr a0
                               movs.w a0,@r5+
 50
                               movx.w a0,@r5+
                               movx.w a0,@r5+
                               movx.w a0,@r5+
                               movx.w a0,@r5+
            mov r6, r0
 55
            add #1,r0
            rts
            nop
 60
             .align
```

vs\_pos:

```
a0,r2
            sts
                  r2, r3
            mov
      add
            \#-1,r3
            add
                  r2,r3
  5
                  r1, r3
      cmp/ge
            bf
                   vsloop5
                   r2, r1
      cmp/ge
 10
                   vs end2
            bt
      ; Case 5: maximum positive value still has room for normalization
             .align
      vsloop61:
 15
             shal r1
             cmp/ge
                         r2, r1
            bf/s vsloop61
            add #1,r0
      vs_end2:
 20
                   r0,y0
             lds
             psha #16,y0
movs.w @r4+,x1
ij
             psha x1,y0,a0
                               movx.w @r4+,x1
H
                                movx.w @r4+,x1
             psha x1,y0,a1
                                movs.w a0,@r5+
25
                                movx.w a1,@r5+
112
                                movx.w @r4+,x1
             psha x1,y0,a0
٠,١
                                movx.w @r4+,x1
             psha x1, y0, a1
Ħ
                                movx.w a0,@r5+
30
             psha x1,y0,a0
                                movx.w a1,@r5+
100
                                movx.w a0,@r5+
             rts
35
             nop
; Case 4: maximum positive value exceeds maximum range
       vsloop5
                          r1, r3
             cmp/ge
  40
             bt
                   vs end2
             .align
       vsloop5:
             shar rl
  45
             cmp/qe
                          r1, r3
             bf/s vsloop51
             add
                   #-1,r0
             bra
                   vs_end2
  50
             nop
       The following is an algorithm in accordance with a first embodiment of the
       present invention.
  55
       ; search for minimum NLS
                                movs.w @r4+,x0
             pdmsb x0,a0
                                movx.w @r4+,x0
             pdmsb x0,y0
  60
             pcmp a0,y0
                                movx.w @r4+,x0
         dct pcopy y0,a0
```

```
pdmsb x0,y0
            pcmp a0,y0
                              movx.w @r4+,x0
        dct pcopy y0,a0
            pdmsb x0,y0
 5
            pcmp a0,y0
                                     movx.w @r4,x0
                 pcopy y0,a0
        dct
                              movx.w @r4+r8,x1;dummy movx to reset r4=&IN[0]
            pdmsb x0,y0
                              movx.w @r4+r8,x1
            pcmp a0,y0
                              movx.w @r4+r8,x1
        dct pcopy y0,a0
                              movx.w @r4+r8,x1
10
            psha #-16,a0
                                     ;r0=NLS MIN
                  a0, r0
            sts
      ;Case 1: zero input vector
15
                     #31, r0
            cmp/eq
            bf/s VSCALE1_check_NLSeq31_end
                  r6, r7
                             ;r6=MLS
            mov
                  r6, r0
            mov
                                     ; set r0=NLS = MLS + 1
                  #1, r0
20
            add
            pclr a0
                               movs.w a0,@r5+
Œ
                               movx.w a0,@r5+
movx.w a0,@r5+
                               movx.w a0,@r5+
25
                               movx.w a0,@r5+
11.5
            rts
١...[
            nop
Ħ
      ;Case 2: non-zero input vector
      VSCALE1 check_NLSeq31_end:
17
                  #<del>-</del>14, r7
                               ;r7=MLS-14
            add
171
                               ;r0=NLS = NLSmin + (MLS-14)
            add
                  r7, r0
125
            lds
                  r0, y0
135
12
      psha #16,y0
                               movs.w @r4+,x0
===
                               movx.w @r4+,x1
            psha x0,y0,a0
                               movx.w @r4+,x0
            psha x1,y0,a1
                                movs.w a0,@r5+
 40
                                movx.w a1,@r5+
             psha x0,y0,a0
                                movx.w a0,@r5+
                                movx.w @r4+,x1
                                movx.w @r4+,x0
             psha x1,y0,a1
 45
                                movx.w a1,@r5+
             psha x0,y0,a0
                                movx.w a0,@r5+
             rts
             nop
 50
```